Sum Product Networks for Activity Recognition

Mohamed R. Amer, *Member, IEEE,* and Sinisa Todorovic, *Senior Member, IEEE*

Abstract—This paper addresses detection and localization of human activities in videos. We focus on activities that may have variable spatiotemporal arrangements of parts, and numbers of actors. Such activities are represented by a Sum-Product Network (SPN). A product node in SPN represents a particular arrangement of parts, and a sum node represents alternative arrangements. The sums and products are hierarchically organized, and grounded onto space-time windows covering the video. The windows provide evidence about the activity classes based on the Counting Grid (CG) model of visual words. This evidence is propagated bottom-up and top-down to parse the SPN graph for the explanation of the video. The node connectivity and model parameters of SPN and CG are jointly learned under two settings, weakly supervised, and supervised. For evaluation, we use our new Volleyball dataset, along with the benchmark datasets VIRAT, UT- Interactions, KTH, and TRECVID MED 2011. Our video classification and activity localization are superior to those of the state of the art on these datasets.

Index Terms—Sum-Product Networks, Activity Recognition, Hierarchical Models

1 Introduction

This work addresses the problem of activity detection and localization in videos. Our focus is on activities with multiple, alternative structures. An activity structure is defined as a spatiotemporal arrangement of subactivities, where the recursion ends at primitive actions. Variations in activity structure may arise from different sets of primitive actions, and their different space-time arrangements. For example, according to the rules of volleyball, many alternative sequences of ball passes between the players may all define a unique volleyball play, e.g., "setting-ball-to-left". This problem arises in many applications.

A common approach to representing an activity structure is to explicitly model temporal relations (e.g., "followed-by", "before", "during") between subactivities. Such expressive models typically have intractable inference and learning. In this work, we present an expressive activity model with efficient exact inference. Our key intuition is that changes in configurations of primitive actions, comprising an activity, will give rise to changes in histograms of video features. For example, a change in feature histograms will occur when "walking" is followed by "jumping". Instead of explicitly capturing temporal configurations of primitives, we will model changes of feature histograms across the video, characterizing the activities. For modeling and computational efficiency, we will explicitly capture hierarchical changes of feature histograms, so as to account for: i) Hierarchical activity decompositions into subactivities until primitive actions, and ii) Sharing of primitives among a number of more complex subactivities.

To represent a video, we place a regular, space-time grid of points across the video, as illustrated in Fig. 1a, and partition the video into a set of space-time windows, each encompassing a number of grid points. Every grid point is associated with a visual word from a dictionary. Every window is modeled as a Bag-of-Words (BoW) characterized by a multinomial distribution over counts of visual words occurring within the window. When the BoWs fall on the video foreground, i.e., activity instance, then they are taken to represent activity primitives.

A Naïve-Bayes product of the aforementioned multinomial distributions of BoWs is called a *Counting Grid* model. CG was used for capturing constraints and changes of feature histograms across different parts of a still image [2], or text document [3]. We use CGs to represent space-time configurations of activity primitives. Specifically, the product of multinomial distributions of a set of BoWs extracted from the video are used to represent a particular spatiotemporal arrangement of the activity primitives. For short, we will use the phrase a product of BoWs to indicate the model of such an arrangement of the primitives. For modeling alternative configurations of variable activity structure, these products are combined in a mixture distribution. A mixture of BoW products can be interpreted as a model of subactivity, where each component in the mixture captures the subactivity's variable structure.

Our key contribution is to model an activity by hierarchically organizing the mixtures of BoW products. The mixtures of a given hierarchical level are used as product terms of the Naïve-Bayes model at the next parent level. We formalize this hierarchical model as the sum-product network (SPN) illustrated in Fig. 1b. SPN is a generalized directed acyclic graph, consisting of three types of nodes – namely, terminal, product, and sum nodes [4], [5]. The terminal nodes correspond to the BoWs of the video. The product nodes compute the Naïve-Bayes products of the BoWs. The sum nodes compute the mixture distributions of the Naïve-Bayes products of the BoWs. All children of a sum are products or terminals, and all children of a product are sums. Children nodes in SPN can be reused and shared by multiple distinct parents. Edges connecting a sum node with its children products are weighted,

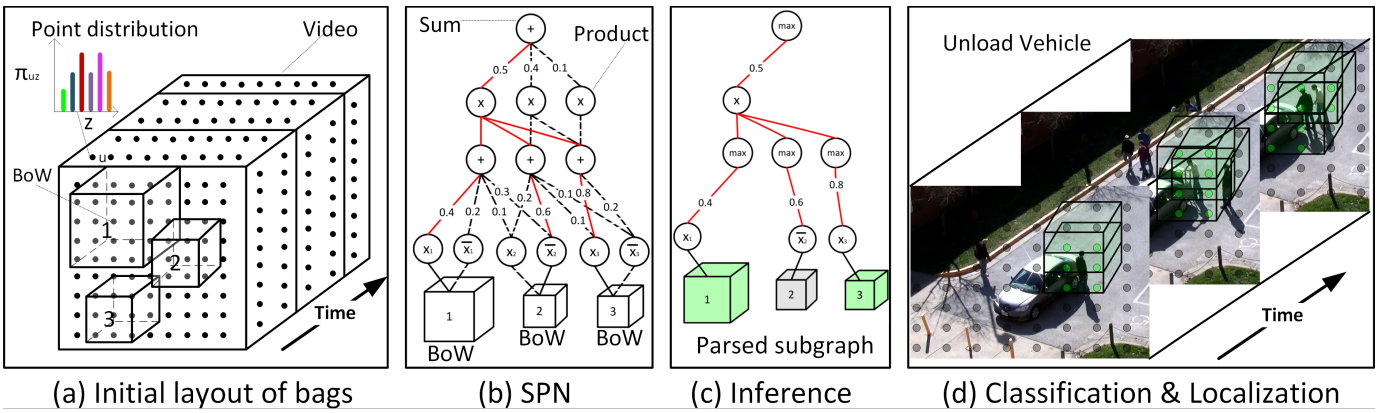


Fig. 1. Our approach: (a) A video is represented by the counting grid model (CG) of visual words; every grid point u is assigned a distribution of word counts π_{uz}; space-time windows B_b are placed across the counting grid and characterized by a Bag-of-Words model in terms of aggregate distributions of word counts on the grid that fall within the window, $\sum_{u \in B_b} \pi_{uz}$. (b) Our activity model is the sum-product network (SPN) that consists of levels of sum and product nodes, ending with space-time windows at terminal nodes; children nodes in the SPN can be shared by multiple parents. (c) SPN inference amounts to parsing, and identifying foreground (green space-time windows). (d) Localization of the activity "unloading of the trunk" in an example sequence from the VIRAT dataset [1].

where the weights correspond to the relative significance of the component in the mixture distribution.

SPN is suitable for capturing alternative structures of an activity, because the product nodes can encode particular configurations of BoWs, i.e., primitives of an activity, whereas the sum nodes can account for alternative configurations. Also, SPN can compactly encode a large number of alternative arrangements of BoWs, because SPN consists of a number of levels of sums and products, where children nodes are shared by parents.

When a new video is encountered, we place a set of space-time windows across the video's regular grid. We characterize the windows with BoWs, and use them as terminal nodes of the SPN. SPN inference amounts to parsing the SPN graph, i.e., selecting a subset of optimal sum, product, and terminal nodes (i.e., BoWs) that yields the explanation of activity occurrence Fig. 1c. The resulting parse is a tree. The video is assigned the label of the activity whose parse tree yields the highest parse score. The selected subset of space-time windows localize foreground video parts Fig. 1d.

For our evaluation, we have compiled and annotated a new Volleyball dataset. Our video classification and activity localization are superior to those of the state of the art on the benchmarks datasets, including VIRAT [1], UT-Interactions [6], KTH [7], TRECVID MED 2011 [8], and Volleyball [9].

Contributions:

- Activity representation that integrates SPN+CG in a unified framework;
- Introducing new hidden random variables over the graph connectivity of our SPN+CG model, whereas previous approaches on SPN including our preliminary work treat SPN edges deterministically.
- Bottom-up and top-down inference algorithm;
- Joint learning of SPN and CG under both weak supervision, and supervision
- Volleyball dataset.

In the following, Sec. 2 reviews prior work; Sec. 3 specifies SPN; Sec. 5 formulates CG, and a joint model of SPN and CG; Sections 6 and 7 specify our inference and learning algorithms; and Sec. 8 presents our experiments.

2 Prior Work

Our literature review is focused on prior work that models activities using graphical models. Then, we relate our work to that on aggregating counts of visual words in the video, and non-linear deep models. Finally, we present our contributions, and explain what is new in this paper relative to our preliminary work of [9].

Graphical models have been successfully used for modeling spatiotemporal structure of activities [10]–[12]. Representative models include Dynamic Bayesian Networks [13]–[15], hierarchical graphical models [16]–[18], AND-OR graphs [19]–[21], and Logic Networks [22]–[24]. Recognition rates increase even further by grounding graphical models onto object-detector responses [16], rather than raw video features (e.g., optical flow). However, graphical models relevant for activity recognition are typically intractable. As learning algorithms use inference to estimate latent variables on training data under weak supervision, and generally assume exact inference, their behavior in the context of heuristic inference is not well understood. In contrast, SPN allows exact inference under certain conditions that are unrestrictive for our purposes, as discussed later.

Among the above graphical models, SPNs are most related to AND-OR graphs, which are also capable of encoding alternative decompositions and configurations of activities [19]–[21], [25]–[27]. AND-OR graphs typically require a manual specification of nodes and graph connectivity, each associated with hand-picked semantic meaning (with few exceptions [28]). Thus, learning AND-OR graphs usually amounts to only learning model parameters of nodes and edges representing user-specified activity parts. In contrast, SPN has a "deeper"

graph with significantly more levels than in AND-OR graph, where the levels, nodes, and connectivity are jointly estimated along with model parameters under weak supervision. As a result, SPN automatically identifies on its own latent activity parts relevant for detection and localization, instead of requiring user specification.

On datasets with structure-rich activities (e.g., UT interactions [6]), graphical models have superior performance over representations which do not explicitly capture activity structure, e.g., Bag-of-Words (BoW) [29], [30], and probabilistic Latent Semantic Analysis (pLSA) and Latent Dirichlet Allocation (LDA) [31]. Video representations that aggregate local video features have been used only for single-actor actions [30], [32], [33], punctual and repetitive actions [34], or activities with parts that occur in a fixed temporal order [35].

SPNs can be viewed as probabilistic, general-purpose convolution networks, with max-pooling operations. Convolutional Neural Networks and other non-linear deep architectures have been used for activity recognition [36], [36]–[39]). Deep networks are constructed by stacking a number of levels of graphical models. This reduces their learning to parameter estimation. Unlike deep networks, SPN has linear complexity in the number of nodes, under unrestrictive conditions [4].

Our contributions include a unified model of SPN and CG for activity representation, and *joint* learning of SPN and CG. Learning the connectivity and edge weights of the SPN graph can be formulated as a variational EM algorithm. In particular, we extend SPN learning presented in [4], [5] to include learning of CG. We formulate a new learning algorithm for this joint learning generalized from the discriminative learning framework of [5]. We study the learning algorithm under a weakly supervised setting and a supervised setting.

This paper extends our preliminary work [9] in three aspects. First, in [9], the SPN graph connectivity is treated *deterministically*, whereas we here relax the deterministic constraints on the SPN graph connectivity, and explicitly infer the most probable graph structure of SPN. Second, this paper presents new activity recognition experiments, missing in [9], that are aimed at testing: i) Sensitivity to various input parameters and the number of training examples; ii) Valid vs. invalid graph structure of SPN; iii) Supervised vs. weakly supervised learning of SPN; and iv) Performance on detecting complex events of challenging TRECVID MED 2011 dataset [8].

3 Review of SPN for Binary Problems

This section reviews key concepts of SPN [4], which we will use later to specify our model of human activities. Tab. 2 summarizes our notation. The random variables are denoted with capital letters, and their particular values with small letters.

SPN is a rooted, directed acyclic graph with three types of nodes: terminal, sum, and product nodes, as illustrated in Fig. 1b.

Notation	Explanation
Y	Latent variable of activity classes.
$\boldsymbol{H}=\{H_j\}$	Latent indicators of product nodes.
$\boldsymbol{X}=\{(X_b,\overline{X}_b)\}$	Latent indicators at terminal nodes.
$\mathbf{C}=\{\mathbf{C}_b\}$	Observable counts of visual words.
$\mathcal{B}=\{B_b\}$	Space-time windows.
$\mathcal{Z}=\{z\}$	Dictionary of visual words.
$\boldsymbol{\pi}_u=[\pi_{uz}]$	Codeword distributions at grid points u.
$\theta=[\theta_z]$	Hyper parameters of the CG model
$S(\cdot)$	A score of SPN.
$\boldsymbol{w}=\{w_{ij;y}\}$	Weights of sum-product edges (i,j).

Fig. 2. Table of notation used in Sections 3-7.

The terminal (leaf) nodes represent pairs of indicator random variables $\boldsymbol{X}=\{(X_b,\overline{X}_b): X_b,\overline{X}_b\in\{0,1\},\ b=1,...,n\}$. They are aimed at grounding SPN onto a set of n spatiotemporal windows of the video. Each pair of variables $(X_b,\overline{X}_b)$ indicates if the corresponding bth window belongs to the video foreground where the detected human activity occurs, or to the background. Specifically, the values $(X_b,\overline{X}_b)$=$(1,0)$ and $(X_b,\overline{X}_b)$=$(0,1)$ indicate that bth window belongs to the foreground and background, respectively. For each human activity $y=1,\ldots,A$, we define the probability of X_b given the observable visual cues $\boldsymbol{c}_b$ in the corresponding bth spatiotemporal window, $P_{X_b}(x_b|c_b,y)$. A more detailed specification of $P_{X_b}(x_b|c_b,y)$ is given in the next section. Since in this section we consider SPN for binary problems, for simplicity, we here drop $\boldsymbol{c}_b$ and y from notation. We also specify: $P_{\overline{X}_b}(\overline{x}_b)=1-P_{X_b}(\overline{x}_b)=P_{X_b}(1-\overline{x}_b)$.

In inference, the joint probability $P_{\boldsymbol{X}}(\boldsymbol{x})$ is used for localizing the target human activity in the video as $\hat{\boldsymbol{x}}=\arg\max_{\boldsymbol{x}} P_{\boldsymbol{X}}(\boldsymbol{x})$. SPN serves to efficiently compute $P_{\boldsymbol{X}}(\boldsymbol{x})$, and thus efficiently conduct inference. As shown in [4], SPN estimates $P_{\boldsymbol{X}}(\boldsymbol{x})$ as a polynomial of sums and products of $\boldsymbol{x}$. This polynomial is recursively computed using the hierarchy of sum and product nodes in SPN. The sum and product nodes of SPN are arranged in alternating levels forming a hierarchical graph, i.e., all children of a sum node are products or terminals, and all children of a product node are sums. The root node is a sum node, and produces the polynomial called the SPN score, $S(\boldsymbol{x})$, which is used to compute $P_{\boldsymbol{X}}(\boldsymbol{x})$ as:

$$P_{\boldsymbol{X}}(\boldsymbol{x})=S(\boldsymbol{x})/S(\mathbf{1}), \tag{1}$$

where $S(\mathbf{1})$ is the SPN score when all the binary variables are set to 1, $\boldsymbol{X}=\mathbf{1}$. As we will show below, $S(\mathbf{1})$ normalizes $S(\boldsymbol{x})$ in (1), because setting any pair of variables to 1 in $S(\boldsymbol{x})$, e.g., $(X_b=1,\overline{X}_b=1)$, amounts to marginalizing out these two variables from $S(\boldsymbol{x})$.

In the following, we explain how to recursively compute $S(\boldsymbol{x})$. Indices of the sum nodes are i,l; indices of the product nodes are j,k; and i^+ denotes the set of children of i. An edge (i,j) that connects a sum node i with its child node $j\in i^+$ has a non-negative weight $w_{ij}\geq 0$, where $\sum_{j\in i^+} w_{ij}=1$. All edges (k,l) that connect a product node k with its children nodes $l\in k^+$ have weights set to 1. We also use $\boldsymbol{X}_i\subseteq\boldsymbol{X}$ to denote a subset of variables that can be reached from node i

in SPN. Using this notation, we recursively define the scores produced by sum and product nodes as:

$$S_i(\boldsymbol{x}_i) = \sum_{j\in i^+} w_{ij} S_j(\boldsymbol{x}_j), \quad S_k(\boldsymbol{x}_k) = \prod_{l\in k^+} S_l(\boldsymbol{x}_l). \quad (2)$$

The recursion ends at sum nodes that are connected to the terminal nodes, where

$$S_i(\boldsymbol{x}_i) = \sum_{b\in i^+} [w^1_{ib} x_b P_{X_b}(x_b) + w^2_{ib} \overline{x}_b P_{\overline{X}_b}(\overline{x}_b)]. \quad (3)$$

From (3), setting $(X_b = 1, \overline{X}_b = 1)$ amounts to marginalizing out these two variables from $S(\boldsymbol{x})$.

From (2) and (3), SPN can be viewed as a mixture model, whose children are the mixture components, which, in turn, are products of mixture models[1].

Validity: As shown in [4], complexity of computing $S(\mathbf{1})$, defined in (1), is linear in the number of SPN nodes, when the connectivity of SPN graph is valid. An architecture is valid if it is complete and consistent. SPN is complete if for every sum node its children product nodes have access to the same set of terminal nodes as that sum node. SPN is consistent if every product node does not have in its set of children any pair of random variables $(X_b, \overline{X}_b) \in \boldsymbol{X}$. Note that in this work the consistency constraint is always satisfied, by construction, since the product nodes are not directly linked to the terminal nodes. The validity of SPN is not restrictive for our modeling of activity classes. This is because a valid SPN can be defined over a large graph with many redundant nodes and edges. Then, in learning, some model weights w_{ij} can be estimated as zero. Since product nodes j whose corresponding w_{ij} is equal to zero do not affect the SPN score $S(\cdot)$, learning that $w_{ij} = 0$ effectively amounts to "removing" these redundant product nodes and their descendant nodes from the initial, large SPN graph. After learning, the resulting SPN will be valid, if the initial graph is valid[2].

4 Grounding SPN on Counting Grid

This section explains how to compute P_{X_b} used in (3) for computing the score of SPN. The video is partitioned into a set of spatiotemporal windows, $\mathcal{B} = \{B_b : b = 1, 2, ..., n\}$. The binary random variable X_b is associated with every window B_b, and $P_{X_b}(X_b{=}1)$ denotes the probability of selecting B_b as a part of video foreground. P_{X_b} is modeled using CG [2], [3], specified below.

Given a dictionary of visual words, $\mathcal{Z}$, their occurrences in the video are governed by CG. CG is defined over a regular space-time grid of points spanning the video, as illustrated in Fig. 1a. Every point u on the space-time grid probabilistically generates one visual word $z \in \mathcal{Z}$ from the distribution $\boldsymbol{\pi}_u = [\pi_{uz}]$. Each π_{uz} represents the probability of occurrence of word z at point u on the grid, such that $\forall u$, $\sum_{z\in\mathcal{Z}} \pi_{uz} = 1$.

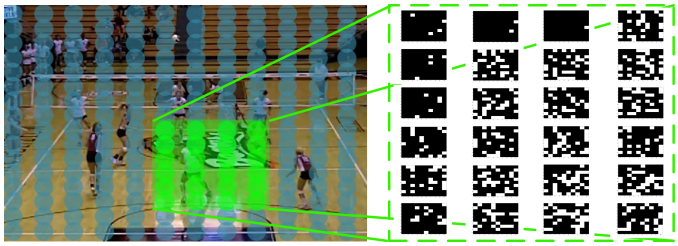


Fig. 3. A frame from our volleyball videos with the space-time grid of points (blue), and a foreground space-time window B (green), and visual words generated by CG within B (right).

Fig. 3 illustrates a frame from our volleyball videos, and the space-time grid of points and a foreground space-time window b overlaid over the frame. The visual words generated by CG within b are also depicted. The words were computed using the method of [36].

We use the point distributions $\boldsymbol{\pi}_u$ to specify a multinomial distribution of counts of visual words, $P_{\mathbf{C}}(\boldsymbol{c}_b)$, for every spatiotemporal window $B_b \in \mathcal{B}$. First, we use the aggregate distribution of word occurrences within the window, $\sum_{u\in B_b} \pi_{uz}$, $\forall z \in \mathcal{Z}$, to probabilistically sample a visual word at every grid point u in B_b. Then, $P_{\mathbf{C}}(\boldsymbol{c}_b)$ is specified as a multinomial distribution of counts of visual words $\boldsymbol{c}_b = [c_{bz}]$ observed within B_b:

$$P_{\mathbf{C}}(\boldsymbol{c}_b|\boldsymbol{\pi}_{B_b}) \propto \prod_z [\sum_{u\in B_b} \pi_{uz}]^{c_{bz}}. \quad (4)$$

We specify $P_{X_b}(X_b = 1|\boldsymbol{c}_b)$ as the posterior probability of observing $\boldsymbol{c}_b$ counts of visual words in B_b

$$P_{X_b}(X_b = 1|\boldsymbol{c}_b) \propto P_{\mathbf{C}}(\boldsymbol{c}_b|\boldsymbol{\pi}_{B_b}) P(\boldsymbol{\pi}_{B_b}) \quad (5)$$

where $P_{\mathbf{C}}(\boldsymbol{c}_b|\boldsymbol{\pi}_{B_b})$ is defined in (4), and $P(\boldsymbol{\pi}_{B_b})$ is the Dirichlet prior with hyper parameters $\boldsymbol{\theta} = [\theta_z]$:

$$P(\boldsymbol{\pi}_{B_b}) \propto \prod_z [\sum_{u\in B_b} \pi_{uz}]^{\theta_z - 1}. \quad (6)$$

In the next section, we will specify a joint model of SPN and CG aimed at detecting and localizing human activities, $y = 1, \ldots, A$ occurring in the video. From (4)–(6), for every activity, we define the probability that B_b belongs to the video foreground of y as

$$P_{X_b}(X_b = 1|\boldsymbol{c}_b, y) \propto \prod_z [\sum_{u\in B_b} \pi_{uz;y}]^{c_{bz}+\theta_{z;y}-1}. \quad (7)$$

The probability that the spatiotemporal window B_b belongs to the background of activity y is computed as $P_{X_b}(X_b = 0|\boldsymbol{c}_b, y) = 1 - P_{X_b}(X_b = 1|\boldsymbol{c}_b, y)$.

5 The Joint Model

This section formulates a joint model of SPN and CG. SPN is grounded via its terminal nodes onto spatiotemporal windows of the video, where visual cues extracted

1. SPN score in Fig. 1b is $S(\boldsymbol{x}) = .5(.4x_1 P_{X_1}(x_1) + .2\overline{x}_1 P_{\overline{X}_1}(\overline{x}_1) + .1x_2 P_{X_2}(x_2) + .3\overline{x}_2 P_{\overline{X}_2}(\overline{x}_2))(.2x_2 P_{X_2}(x_2) + .6\overline{x}_2 P_{\overline{X}_2}(\overline{x}_2) + .1x_3 P_{X_3}(x_3) + .1\overline{x}_3 P_{\overline{X}_3}(\overline{x}_3))(.8x_3 P_{X_3}(x_3) + .2\overline{x}_3 P_{\overline{X}_3}(\overline{x}_3)) + .1(.8x_3 P_{X_3}(x_3) + .2\overline{x}_3 P_{\overline{X}_3}(\overline{x}_3)) + .4(.2x_2 P_{X_2}(x_2) + .6\overline{x}_2 P_{\overline{X}_2}(\overline{x}_2) + .1x_3 P_{X_3}(x_3) + .1\overline{x}_3 P_{\overline{X}_3}(\overline{x}_3))(.8x_3 P_{X_3}(x_3) + .2\overline{x}_3 P_{\overline{X}_3}(\overline{x}_3))$.

2. Note that the SPN model presented in [4] is similar to the And-Or Tree (AOT) model presented in [40]. The main difference is in the SPN's validity constrains that allow for generative and discriminative training compared to the discriminatively trained AOT in [40].

from each window are modeled using CG. For every activity class, SPN computes a polynomial of foreground-background indicators associated with the spatiotemporal windows. The activity class that gives the largest SPN score is taken as present in the video. Inference also produces a parse graph of SPN whose terminal nodes identify the video foreground.

Random variables of the joint model include the following. As observables, we use the same counts of visual words $\boldsymbol{c}_b$ observed within spatiotemporal windows of the video $\mathcal{B} = \{B_b : b = 1, 2, ..., n\}$, as described in Sec. 4. We also use the latent pairs of foreground-background indicators $\boldsymbol{X} = \{(X_b, \overline{X}_b) : b = 1, ..., n\}$ associated with every window $B_b \in \mathcal{B}$, specified in Sec. 3. For activity classes $y = 1, \ldots, A$, the probability of X_b given $\boldsymbol{c}_b$, $P_{X_b}(x_b; y)$, is given by (7). These variables is extended by another set of latent variables, as explained below.

We use a random variable Y to denote the activity class $Y \in \{1, ..., A\}$. We also associate latent indicators $\boldsymbol{H} = \{H_j : H_j \in \{0, 1\}, j = 1, 2, \dots\}$ with every product node j in SPN. When $H_j = 1$ for $j \in i^+$, this means that the sum computed at node i includes as one of its terms the product computed at node j; otherwise, jth product is excluded from the sum. These variables help us relax the validity constraints on the SPN graph connectivity. Specifically, in inference, our goal will be to infer product-node indicators $\hat{\boldsymbol{h}}$ which maximize the SPN score for activity recognition.

We specify a joint distribution, $P_{Y\boldsymbol{HX}}(y, \boldsymbol{h}, \boldsymbol{x})$, that characterizes our joint model as

$$P_{Y\boldsymbol{HX}}(y, \boldsymbol{h}, \boldsymbol{x}) = \frac{S(y, \boldsymbol{h}, \boldsymbol{x})}{\sum_{y=1}^{A} S(y, \mathbf{1}, \mathbf{1})}, \tag{8}$$

where, similar to (1), $S(y, \boldsymbol{h}, \boldsymbol{x})$ is the score produced at the root node of SPN, and $S(y, \mathbf{1}, \mathbf{1})$ is the SPN score when all indicators $\boldsymbol{H}$ and $\boldsymbol{X}$ are marginalized out.

To compute (8), we extend the definitions of the sum and product scores of SPN presented in Sec. 3 as

$$\begin{aligned} S_i(y, \boldsymbol{h}_i, \boldsymbol{x}_i) &= \sum_{j \in i^+} w_{ij;y} h_j S_j(y, \boldsymbol{h}_j, \boldsymbol{x}_j), \ y = 1, \ldots, A, \\ S_k(y, \boldsymbol{h}_k, \boldsymbol{x}_k) &= \prod_{l \in k^+} S_l(y, \boldsymbol{h}_l, \boldsymbol{x}_l), \ y = 1, \ldots, A, \end{aligned} \tag{9}$$

where the model parameters $w_{ij;y}$ are indexed by the activity class y. The recursion ends with the scores of sum nodes that connect to the terminals, specified for every activity $y = 1, \ldots, A$ as

$$\begin{aligned} S_i(y, \boldsymbol{h}_i, \boldsymbol{x}_i) = \sum_{b \in i^+} \Big[& w^1_{ib;y} x_b P_{X_b}(x_b | \boldsymbol{c}_b, y) \\ & + w^2_{ib;y} \overline{x}_b P_{\overline{X}_b}(\overline{x}_b | \boldsymbol{c}_b, y) \Big]. \end{aligned} \tag{10}$$

From (7), (9), (10), our model parameters include $\Omega = \Big\{ \{w_{ij;y}\}, \{\pi_{uz;y}\}, \{\theta_{z;y}\} : y = 1, \ldots, A \Big\}$.

Validity: Similar to the binary SPNs introduced in [4], it is worth noting that our joint model of SPN and CG allows for efficient computation of $S(y, \mathbf{1}, \mathbf{1})$, given by (8). Specifically, its complexity is linear in the number of SPN nodes, when the connectivity of SPN graph is valid, i.e., satisfies the completeness and consistency constraints, defined in Sec. 3. Again, the consistency constraint is always satisfied by construction. The validity constraints of our joint model is not restrictive for modeling human activities that we consider in this work. This is because a valid SPN can be defined over a large graph with many redundant nodes and edges. When latent indicators $\hat{h}_j$ are estimated to be zero, or when weights w_{ij} are learned to have zero values, then the corresponding product nodes j cannot affect the SPN score $S(\cdot)$. That is, these product nodes and their descendants are effectively "removed" from the valid network, yielding a valid joint model of SPN and CG.

6 Inference

Given a video, we recognize activity $\hat{y}$, and identify spatiotemporal windows $\hat{\boldsymbol{x}}$ that belong to foreground. For recognition, we maximize the marginal of the joint distribution, given by (8), as

$$\hat{y} = \arg\max_{y} \sum_{\boldsymbol{h}, \boldsymbol{x}} P_{Y\boldsymbol{HX}}(y, \boldsymbol{h}, \boldsymbol{x}) = \arg\max_{y} S(y, \mathbf{1}, \mathbf{1}), \tag{11}$$

where $S(y, \mathbf{1}, \mathbf{1})$ can be computed in a *bottom-up* pass through the network. For localization, we conduct a greedy *top-down* parsing of SPN by estimating latent indicators $\hat{\boldsymbol{h}}$ at each SPN level. When the top-down parsing reaches the terminal nodes, we estimate the latent foreground indicators $\hat{\boldsymbol{x}}$. Thus, our inference consists of the bottom-up and top-down computational steps.

We extract video features at points of the counting grid, and map the features to a dictionary of visual words. Then, we place a set of windows $\mathcal{B}$ across the grid, and compute the counts of visual words within every bag. These counts are taken as observables for computing the SPN scores $S(y, \boldsymbol{h}, \boldsymbol{x})$, for activity classes $y = 1, \ldots, A$.

Bottom-up pass: The scores of sum and product nodes, given by (9), are recursively computed with all product-node indicators set to 1, $H_j = 1$, $j = 1, 2, ...$, as well as all foreground indicators set to 1, $X_b = \overline{X}_b = 1$, $b = 1, ..., n$. By setting $\boldsymbol{H} = \mathbf{1}$ and $\boldsymbol{X} = \mathbf{1}$, we marginalize out $\boldsymbol{H}$ and $\boldsymbol{X}$ in the SPN score. The bottom-up pass starts at the sum nodes connected to the terminals, where we compute for every activity $y = 1, \ldots, A$:

$$S_i(y, \mathbf{1}, \mathbf{1}) = \sum_{b \in i^+} [w^1_{ib;y} P_{X_b}(1 | \boldsymbol{c}_b, y) + w^2_{ib;y} P_{\overline{X}_b}(1 | \boldsymbol{c}_b, y)]. \tag{12}$$

These scores are propagated bottom-up, for $y = 1, \ldots, A$:

$$\begin{aligned} S_i(y, \mathbf{1}, \mathbf{1}) &= \sum_{j \in i^+} w_{ij;y} S_j(y, \mathbf{1}, \mathbf{1}), \\ S_k(y, \mathbf{1}, \mathbf{1}) &= \prod_{l \in k^+} S_l(y, \mathbf{1}, \mathbf{1}). \end{aligned} \tag{13}$$

At the root node, the bottom-up pass performs activity recognition as in (11).

Top-down pass: Our goal is to infer $\hat{\boldsymbol{x}}$ for the recognized $\hat{y}$. These estimates are computed top down, via estimating $\hat{\boldsymbol{h}}$ for each SPN level. Specifically, we parse the SPN graph, from the root node down, by identifying $\hat{\boldsymbol{h}}$ at every sum node i from the following linear integer problem:

$$\begin{aligned} &\max_{\{h_j:j\in i^+\}} \sum_{j\in i^+} h_j[w_{ij;\hat{y}} S_j(\hat{y},\mathbf{1},\mathbf{1})] \\ &\text{s.t. } \textstyle\sum_{j\in i^+} h_j = 1; \quad h_j\in\{0,1\}, \text{ for all } j\in i^+. \end{aligned} \tag{14}$$

The entire subgraphs rooted at product nodes j for which we estimated $\hat{h}_j = 0$ in (14) are removed from the parse graph. That is, in the greedy top-down pass, the optimization in (14) is computed only for sum nodes whose parent product nodes k got selected $\hat{h}_k = 1$.

The top-down pass ends at sum nodes i that are included in the parse graph and connected to the terminals. In this final step, we estimate the foreground-background video segmentation from the following linear integer problem:

$$\begin{aligned} &\max_{\{x_b,\overline{x}_b\}} \sum_b \Big[x_b,\overline{x}_b\Big]^\top \Big[w^1_{ib;\hat{y}} P_{X_b}(1|\boldsymbol{c}_b,\hat{y}), w^2_{ib;\hat{y}} P_{\overline{X}_b}(1|\boldsymbol{c}_b,\hat{y})\Big] \\ &\text{s.t. } x_b+\overline{x}_b = 1,\ x_b,\overline{x}_b\in\{0,1\}, \text{ for all } b\in i^+. \end{aligned} \tag{15}$$

Our inference is summarized in Alg. 1.

Algorithm 1: Inference

Input: Observed counts of visual words in the video. Parameters $\Big\{\{w_{ij;y}\},\{\pi_{uz;y}\},\{\theta_{z;y}\} : y=1,\dots,A\Big\}$.

Output: Activity class $\hat{y}$, parse $\hat{\boldsymbol{h}}$ and foreground BoWs $\hat{\boldsymbol{x}}$.

Bottom-Up Pass:
Compute $\{P_{X_b}(1|\boldsymbol{c}_b,y) : b=1,...,n,\ y=1,...,A\}$ as in (7);
Compute $\{S_i(y,\mathbf{1},\mathbf{1}) : y=1,...,A\}$ for sum nodes i at the leaf level of SPN as in (3);
Compute for all sum nodes $\{S_i(y,\mathbf{1},\mathbf{1})\}$ and all product nodes $\{S_k(y,\mathbf{1},\mathbf{1})\}$ for $y=1,...,A$ as in (13);
Top-Down Pass:
Recognize activity $\hat{y}$ at the root node as in (11);
Identify the parse graph $\hat{\boldsymbol{h}}$ as in (14) ;
Estimate foreground $\hat{\boldsymbol{x}}$ as in (15);

7 LEARNING

One of our key contributions is a *joint* learning of SPN and CG. We extend SPN learning presented in [4], [5] to include learning of CG. We formulate a new variational learning algorithm for this joint learning, using two settings, one is weakly supervised, and the other is supervised. In the weakly supervised setting, we assume that training videos are labeled with activity classes. In the supervised setting, we assume access to ground-truth foreground annotations in addition to class labels of training videos.

Given a set of training videos $\mathcal{T}$ showing only *positive* instances of an activity class, $y = 1,\dots,A$, our goal is to learn parameters $\Omega_y = \{\boldsymbol{w}_y, \boldsymbol{\pi}_y, \boldsymbol{\theta}_y\}$. We assume that all training videos $t \in \mathcal{T}$ are partitioned into the same layout of spatiotemporal windows $\mathcal{B}^t$, and that we can observe counts of visual words within every window. Learning is formalized as maximizing the model's joint distribution over all training videos, $\hat{\Omega}_y = \arg\max_{\Omega_y} \sum_{t\in\mathcal{T}} \log \sum_{\boldsymbol{h}^t,\boldsymbol{x}^t} P_{Y\boldsymbol{H}\boldsymbol{X}}(y,\boldsymbol{h}^t,\boldsymbol{x}^t)$, where y is the ground-truth activity class of training videos in $\mathcal{T}$. From (8), we have

$$\begin{aligned} \hat{\Omega}_y &= \arg\max_{\Omega_y} \sum_{t\in\mathcal{T}} \log \sum_{\boldsymbol{h}^t,\boldsymbol{x}^t} \frac{S(y,\boldsymbol{h}^t,\boldsymbol{x}^t)}{\sum_y S(y,\mathbf{1}^t,\mathbf{1}^t)}, \\ &\approx \arg\max_{\Omega_y} \sum_{t\in\mathcal{T}} \max_{\boldsymbol{h}^t,\boldsymbol{x}^t} \log \frac{S(y,\boldsymbol{h}^t,\boldsymbol{x}^t)}{\sum_y S(y,\mathbf{1}^t,\mathbf{1}^t)}, \\ &\approx \arg\max_{\Omega_y} \sum_{t\in\mathcal{T}} \log \frac{S(y,\hat{\boldsymbol{h}}^t,\hat{\boldsymbol{x}}^t)}{\sum_y S(y,\mathbf{1}^t,\mathbf{1}^t)}, \text{ (weak supervision)} \end{aligned} \tag{16}$$

where, in the weak supervision setting, $\hat{\boldsymbol{h}}^t$ and $\hat{\boldsymbol{x}}^t$ are estimated for each training video $t \in \mathcal{T}$ in the top-down parse as in (14) and (15), respectively. In the supervised setting, instead of $\hat{\boldsymbol{x}}^t$, we use the ground-truth foreground annotation $\boldsymbol{x}^t$ in (16).

For estimating $\hat{\Omega}_y$ in (16) we use iteration which alternates the following two steps:

1) Estimation of $\hat{\boldsymbol{w}}_y^{(\tau+1)}$, given $\boldsymbol{\pi}_y^{(\tau)}$ and $\boldsymbol{\theta}_y^{(\tau)}$, using gradient ascent

$$w^{(\tau+1)}_{ij;y} = w^{(\tau)}_{ij;y} + \eta \Delta w^{(\tau)}_{ij;y}, \tag{17}$$

where $\eta > 0$ is the learning rate, and $\Delta w^{(\tau)}_{ij;y}$ is a gradient of the objective in (16). After updating all weights $\{w_{ij;y} : j \in i^+\}$, as in (17), they are normalized such that they sum to 1, for all $j \in i^+ : w^{(\tau+1)}_{ij;y} = w^{(\tau+1)}_{ij;y} / \sum_{j'\in i^+} w^{(\tau+1)}_{ij';y}$.

2) Estimation of $\hat{\boldsymbol{\pi}}_y^{(\tau+1)}$ and $\hat{\boldsymbol{\theta}}_y^{(\tau+1)}$, given $\hat{\boldsymbol{w}}_y^{(\tau+1)}$, using variational approximation.

In the following, we first specify the gradient $\Delta w^{(\tau)}_{ij;y}$ in the weakly supervised and supervised settings, and then specify the second step of our iterative learning.

7.1 Estimation of the Gradient of SPN Parameters

To estimate the gradient $\Delta w^{(\tau)}_{ij;y}$ from (16), in the weakly supervised setting, we compute $S(y,\hat{\boldsymbol{h}}^t,\hat{\boldsymbol{x}}^t)$ by parsing the SPN graph as in (14). From (14), each sum node selects only a *single* product node to be included in the parse graph. As a result, the SPN score has the form of a product $S(y,\hat{\boldsymbol{h}}^t,\hat{\boldsymbol{x}}^t) = \prod_j [w_{ij;y} S_j(y,\mathbf{1},\hat{\boldsymbol{x}}^t_j)]^{\hat{h}^t_j}$, where $\hat{h}^t_j \in \{0,1\}$ for every product node j in SPN. From (16), it follows that:

$$\begin{aligned} \Delta w_{ij;y} &= \sum_{t\in\mathcal{T}} \frac{\partial\Big[\hat{h}^t_j \log[w_{ij;y} S_j(y,\mathbf{1}^t,\hat{\boldsymbol{x}}^t_j)] - \log S(y,\mathbf{1}^t,\mathbf{1}^t)\Big]}{\partial w_{ij;y}}, \\ &= \frac{N^{\text{WS}}_{ij;y} - |\mathcal{T}|}{w_{ij;y}}, \end{aligned} \tag{18}$$

where $N^{\text{WS}}_{ij;y}$ is the number of times the edge (i,j) is included in the parse graph of SPN using (14) across all training videos in $\mathcal{T}$ for activity class y.

Similar to the derivation in (18), we compute the gradient $\Delta w^{(\tau)}_{ij;y}$ in the supervised setting as

$$\Delta w_{ij;y} \frac{N^{\text{S}}_{ij;y} - |\mathcal{T}|}{w_{ij;y}}, \tag{19}$$

where $N^{\text{S}}_{ij;y}$ is the number of times the edge (i,j) is included in the parse graph of SPN in the supervised setting for training videos of activity y in $\mathcal{T}$.

7.2 Variational Learning of CG

In the weakly supervised setting, the second step of our iterative learning computes $\boldsymbol{\pi}_y$ and $\boldsymbol{\theta}_y$ by using the estimates of SPN weights $\boldsymbol{w}_y$ and foreground-background estimates $\hat{\boldsymbol{x}}^t = \{(\hat{x}^t_b, \hat{\bar{x}}^t_b) : b = 1, \ldots, n\}$, for training videos $t \in \mathcal{T}$ of activity class y. Since $S(\mathbf{1}, \mathbf{1}^t, \mathbf{1}^t)$ does not depend on $\boldsymbol{\pi}_y$ and $\boldsymbol{\theta}_y$, from (16), we have:

$$\begin{aligned}[\hat{\boldsymbol{\pi}}_y, \hat{\boldsymbol{\theta}}_y] &= \max_{\boldsymbol{\pi}_y, \boldsymbol{\theta}_y} \sum_{t \in \mathcal{T}} \log S(y, \hat{\boldsymbol{h}}^t, \hat{\boldsymbol{x}}^t), \\ &= \max_{\boldsymbol{\pi}_y, \boldsymbol{\theta}_y} \sum_{t \in \mathcal{T}} \sum_{B_b \in F^t} \log P_{\mathbf{C}}(\boldsymbol{c}^t_b | \boldsymbol{\pi}_{B_b;y}) P(\boldsymbol{\pi}_{B_b;y}),\end{aligned} \tag{20}$$

where F^t denotes the set of spatiotemporal windows B_b in the parse graph of video t estimated as belonging to foreground, $F^t = \{B_b : B_b \in \mathcal{B}^t, \hat{x}^t_b = 1\}$.

To solve the optimization problem in (20), we find $\hat{\boldsymbol{\pi}}_y, \hat{\boldsymbol{\theta}}_y$ by maximizing a lower variational bound $V(\boldsymbol{\pi}_y, \boldsymbol{\theta}_y)$ defined as follows. The objective in (20) can be lower-bounded by the following expression:

$$\begin{aligned}\tilde{V}(\boldsymbol{\pi}_y, \boldsymbol{\theta}_y) = \sum_{t \in \mathcal{T}} \Big[\sum_{b=1}^{n} Q_{b;y} \log[(\hat{x}^t_b - \hat{\bar{x}}^t_b)/Q_{b;y}] \\ + \sum_{b=1}^{n} Q_{b;y} \sum_{z} (c^t_{bz;y} + \theta_{z;y} - 1) \log \sum_{u \in B_b} \pi_{uz;y} \Big],\end{aligned} \tag{21}$$

where $\{Q_{b;y} : b = 1, ..., n\}$ is the variational distribution over the latent selection of foreground spatiotemporal windows $B_b \in \mathcal{B}$ in the video. Note that the second term of (21) requires computing the summation $\sum_{u \in B_b} \pi_{uz;y}$ over all points of the counting grid within window B_b, before applying the logarithm. We bound this second term by using the Jensen's inequality as:

$$\log \sum_{u \in B_b} \pi_{uz;y} = \log \sum_{u \in B_b} \frac{\pi_{uz;y} r_{uz;y}}{r_{uz;y}} \geq \sum_{u \in B_b} r_{uz;y} \log \frac{\pi_{uz;y}}{r_{uz;y}} \tag{22}$$

where $r_{uz;y}$ is a probability distribution over points u, $r_{uz;y} \geq 0$, and $\sum_{u \in B_b} r_{uz;y} = 1$. $r_{uz;y}$ can be computed as a function of $\pi_{uz;y}$, by constrained optimization of the objective in (22), resulting in $r_{uz;y} = \frac{\pi_{uz;y}}{\sum_{u \in B_b} \pi_{uz;y}}$. By substituting the expression of (22) in (21), we derive our variational bound of the objective in (20) as

$$\begin{aligned}V(\boldsymbol{\pi}_y, \boldsymbol{\theta}_y) = \sum_{t \in \mathcal{T}} \Big[\sum_{b=1}^{n} Q_{b;y} \log[(\hat{x}^t_b - \hat{\bar{x}}^t_b)/Q_{b;y}] \\ + \sum_{b=1}^{n} Q_{b;y} \Big[\sum_{z} (c^t_{bz} + \theta_{z;y} - 1) \sum_{u \in B_b} r_{uz;y} \log(\frac{\pi_{uz;y}}{r_{uz;y}}) \\ + \sum_{u \in B_b} \lambda_{\pi_u} (\sum_{z} \pi_{uz;y} - 1) \Big] \Big] - \lambda_\theta (\sum_{z} \theta_{z;y} - 1),\end{aligned} \tag{23}$$

where the last two terms account for the following probability-distribution constraints of π_{uz} and θ_z: $\sum_z \pi_{uz} = 1$, and $\sum_z \theta_z = 1$, and $\{\lambda_{\pi_u}\}$ and λ_θ are the Lagrangian coefficients.

Maximizing (23) with respect to $\boldsymbol{\pi}_y$ and $\boldsymbol{\theta}_y$ gives the following update rules:

$$\begin{aligned}\pi^{(\tau+1)}_{uz;y} &\propto \pi^{(\tau)}_{uz;y} \sum_{t \in \mathcal{T}} \sum_{b=1}^{n} \frac{Q^{(\tau)}_{b;y} (c^t_{bz} + \theta^{(\tau)}_z - 1)}{\sum_{u \in B_b} \pi^{(\tau)}_{uz;y}}, \\ \theta^{(\tau+1)}_{z;y} &\propto \theta^{(\tau)}_z \sum_{b=1}^{n} Q^{(\tau)}_{b;y} \sum_{u \in B_b} \frac{\pi^{(\tau)}_{uz;y} \log[\sum_{u \in B_b} \pi^{(\tau)}_{uz;y}]}{\sum_{u \in B_b} \pi^{(\tau)}_{uz;y}}.\end{aligned} \tag{24}$$

where $Q_{b;y}$ that maximizes $V(\boldsymbol{\pi}_y, \boldsymbol{\theta}_y)$ can be computed from (23) as

$$\begin{aligned}&Q^{(\tau+1)}_{b;y} \propto \\ &\exp[\sum_{t,z,i \in j+} (w^1_{ib} \hat{x}^t_b - w^2_{ib} \hat{\bar{x}}^t_b)(c^t_{bz} + \theta^{(\tau)}_{z;y} - 1) \log \sum_{u \in B_b} \pi^{(\tau)}_{uz;y}].\end{aligned} \tag{25}$$

In the supervised setting, the update equations of $\pi^{(\tau+1)}_{uz;y}$ and $\theta^{(\tau+1)}_{z;y}$, given by (24), remain the same, while $Q^{(\tau+1)}_{b;y}$ is computed by using the ground-truth foreground annotations x^t_b and $\bar{x}^t_b$ in (25), instead of the estimates $\hat{x}^t_b$ and $\hat{\bar{x}}^t_b$.

7.3 Initialization of Learning

The initial distributions $\{\pi^{(0)}_{uz;y}\}$ and $\{\theta^{(0)}_{z;y}\}$ are estimated as the average counts of visual-word occurrences, observed at every point u of the counting grids of training videos $t \in \mathcal{T}$, such that $\sum_z \pi^{(0)}_{uz;y} = 1$, and $\sum_z \theta^{(0)}_{z;y} = 1$.

Based on our empirical evaluation explained in Sec. 8, the initial *valid* graph structure of SPN for learning is specified as follows. The initial height of SPN is set to 8 levels including: the alternating 3 levels of sum nodes and 3 levels of product nodes, the lowest level of terminal nodes, and a single root node at the top. The initial width of SPN at each of the non-terminal alternating levels is set to 10 nodes. At the lowest level of sum nodes, we connect each sum to a distinct group of terminal nodes, i.e., space-time windows in the video. In particular, the video se partitioned into 10 non-overlapping intervals in time, and all binary indicators X_b and $\overline{X}_b$ of space-time windows that fall within interval i are connected to the corresponding sum node i, $i = 1, \ldots, 10$. For the higher levels of SPN, we establish *full* connectivity between all nodes at the consecutive levels, i.e., each parent node has 10 children,

and each child node has 10 parents. Such a structure of SPN is suitable for modeling a range of activity classes, where foreground time intervals in the video are picked by a few lowest-level sum nodes, and their configuration by a few product nodes in the SPN's parse graph.

In our experiments, we also evaluate our performance when the initial graph structure of SPN does not satisfy the completeness constraint, defined in Sec. 3. While there are numerous possibilities to construct an invalid graph structure, for comparison, we use a network as similar to the above valid SPN as possible but with a "nearly full" network connectivity. Thus, the initial height and width of non-terminal levels of SPN are the same as above. Also, the connectivity between the lowest level of sum nodes and terminal nodes is the same as above. But for the higher levels, every parent gets linked to a distinct subset of 9 children nodes. Note that in this way two children product nodes of a parent sum node will have access to different subsets of terminal nodes, which will violate the completeness constraint. At the same time, the nearly full connectivity between the levels of SPN enables sufficient redundancy for learning to "remove" spurious nodes.

Our weakly supervised learning is summarized in Alg. 2. In our experiments, Alg. 2 converges in about $\tau_{\max} = 10 - 20$ iterations, depending on the activity class. After learning, the final SPN graph is obtained by pruning edges with zero weights, and recursively removing non-root parentless nodes.

Algorithm 2: Weakly Supervised Learning

Input: Training videos $\mathcal{T} = \{t\}$ of activity class y
Output: $\Omega_y = \{\boldsymbol{w}_y, \boldsymbol{\pi}_y, \boldsymbol{\theta}_y\}$

```
Initialize the valid SPN graph structure;
Estimate w_y^(0), π_y^(0), θ_y^(0)
for τ = 1 : τ_max do
    for t ∈ T do
        Parse t using Alg. 1, and estimate x̂^t;
    end
    Compute w_y^(τ) using (17) and (18) on T;
    Normalize w^(τ) to sum to 1 for every sum node i ;
    Compute π^(τ) and θ^(τ) as in (24)
end
```

8 RESULTS

This section specifies the datasets used for evaluation, our features, and baselines, as well as presents our quantitative and qualitative results.

8.1 Datasets

Most existing benchmark datasets – e.g., the Weizmann, Trecvid, PETS04, CAVIAR, IXMAS, Hollywood datasets – show relatively simple punctual or repetitive actions with little structural variability [10]–[12]. The Olympic Sports and UCF-101 are more challenging benchmark datasets with structured, single person's activities. However, these datasets are also not suitable for our evaluation, since most of their activity classes have deterministic temporal structure as regulated by the strict specifications of the Olympics or other sports. Surveillance datasets – e.g., the VIRAT Aerial and CLIF datasets – are also not appropriate for our evaluation, since they are recorded from a high altitude where people are poorly visible. The UCLA Courtyard and Collective Activity datasets show group activities which lack temporal structure that our model is specifically aimed at.

For evaluation, we use five datasets: VIRAT 1.0 & 2.0 [1], UT-Interactions [41], KTH [7], 2011 TRECVID Multimedia Event Detection [8], and our own new Volleyball dataset [42].

VIRAT 1.0 & 2.0 includes 11 activity classes: loading and unloading a vehicle, opening and closing the trunk of a vehicle, getting into and out of a vehicle, entering and exiting a facility, running, carrying an object, and gesturing. VIRAT is suitable for our evaluation, because its videos show activities with structural variations. As in [43], we have partitioned the VIRAT footage into shorter clips. Each clip shows an instance of one of the 11 activities, which may take between 5-10s, as well as 10 additional seconds of the original longer video. These additional 10s may occur randomly before and/or after the target activity instance. In this way, we have compiled a dataset with 20 VIRAT clips for each activity, where 50% are used for training, and 50%, for testing.

UT-Interactions consists of 120 video clips, with the standard split of 20% training and 80% testing video, where each clip shows an instance from the set of 6 activity classes: shaking-hands, pointing, hugging, pushing, kicking, and punching.

KTH consists of 2391 short sequences showing 6 types of human actions: walking, jogging, running, boxing, hand waving, and hand clapping. Although our focus is on complex activities, we use KTH for comparison against the state of the art [29], [31], [41], [44] under their setup: videos of 16 subjects are used for training, and videos of other 9 subjects are used for testing. This dataset is a sanitized dataset where all competing approaches have an accuracy of 95% or higher.

TRECVID consists of Internet videos showing 15 events. The videos are split into two sets: DEV-T and DEV-O. The DEV-T set consists of five events: "Board trick" (BT), "Feeding an animal" (FA), "Landing a fish" (LF), "Wedding ceremony" (WC), and "Wood project" (WP). The DEV-O set consists of 10 events: "Birthday party" (BP), "Changing a tire" (CT), "Flashmob gathering" (FG), "Getting a vehicle unstuck" (VU), "Grooming an animal" (GA), "Making a sandwich" (MS), "Parade" (PA), "Parkour" (PR), "Repairing an appliance" (RA), and "Sewing project" (SP). For each event, there are approximately 150 training videos. For testing, there are 10,723 videos in DEV-T, and 32,061 videos in DEV-O with a total duration of 1200 hours. We consider the two test sets separately per the evaluation instructions in [8].

TRECVID is more suitable for evaluating video retrieval, since both DEV-T and DEV-O contain additional null videos that do not show any of the above 15 events. Our performance on these null videos amounts to chance classification, which reduces our average classification accuracy. Nevertheless, we use this dataset for its complexity and comparison with related approaches.

Our **Volleyball dataset** [42] consists of 240 videos showing 6 classes of volleyball set types – namely, setting the ball to the left, right, middle, back right, back left, and back middle Fig. 7. Volleyball dataset is suitable for our purposes, since each activity class can be realized through a wide range of different sequences of ball passes between the volleyball players. There are 40 videos per class, split into 20 training and 20 test videos. Each video is about 4-5 seconds long, and shows one instance of the volleyball set type at resolution 720×480. The volleyball game is a very dynamic group sport, and a 4-second video footage may show the entire volleyball play involving quick player interactions and ball passes. The videos show a large variability of players' movements under occlusion. We provide ground-truth annotations in terms of bounding boxes around the volleyball players engaged in the activity. Our Volleyball dataset is challenging due to very small inter-class differences.

Volleyball, TRECVID, VIRAT and UT-Interactions provide ground-truth foreground annotations, which we use for evaluation of foreground localization.

8.2 Features

For feature extraction, we use the two-layered Stacked Convolutional ISA network (SCISA), recently presented in [36]. We closely follow their setup, since it was thoroughly evaluated and demonstrated as superior to alternative methods. Specifically, we represent a video by a space-time grid of $N \times N \times N$ points that are evenly distributed along each of the spatial and time axes, where $N = 64$ is default. Then, a space-time patch of size (16 pixels) $\times$ (16 pixels) $\times$ (10 frames) centered at each grid point is input to the SCISA network to produce a 500-dimensional feature vector. This feature vector is then mapped to the closest visual word in a 300-dimensional dictionary. As in [36], we train the SCISA network on 200K video patches, and learn the dictionary of visual words on the resulting local features using the K-means, with K=300.

8.3 Baselines

To evaluate sensitivity to our design choices, we define a default variant of our approach, SPN+CG, and compare it to variants SPN+CG+Cubes, SPN+LR, and CG.

SPN+CG uses the counting grid with $N \times N \times N$ points, where $N = 64$. Every grid point is a center of a hierarchy of windows enclosing $(2^{-m}N) \times (2^{-m}N) \times (2^{-m}N)$ neighboring points, where $m = 1, 2, ..., \log_2 \frac{N}{4}$. At each grid point, SPN+CG extracts local features using the SCISA network of [36], and maps them to 300 visual words. Next, we place space-time windows centered at every point of the counting grid. The windows enclose $\frac{N}{2^m} \times \frac{N}{2^m} \times \frac{N}{2^m}$ neighboring points (except windows on the video boundary), where $m = 2, 3, ..., \log_2 \frac{N}{4}$. We consider two strategies for specifying the window size: (a) m is set to one specific value and applied to all windows; (b) m is varied in the interval $[2, \log_2 \frac{N}{4}]$ producing a hierarchy of windows. The SPN network structure is initialized as described in Sec. 7.3. The video is partitioned into 10 time intervals, which are then represented by 10 sum nodes at the lowest level in SPN. Each sum node of the lowest level is connected to all space-time windows that fall within the corresponding time interval. We evaluate SPN+CG when the SPN graph structure is valid, SPN+CG(V), and when the SPN graph structure is not complete, SPN+CG(I). If the type of graph structure is not explicitly stated, then we will assume the valid structure. Finally, SPN+CG is evaluated when the model parameters are learned under weak supervision, SPN+CG(WS), and supervision, SPN+CG(S). For simplicity, we use shorthand notation SPN+CG to denote SPN+CG(WS,V) as our default.

SPN+CG+Cubes changes the feature extraction step, and uses the cuboid spatiotemporal features of [45], which were also used in [31], [41]. A comparison of SPN+CG and SPN+CG+Cubes tests our dependency on the type of local features used.

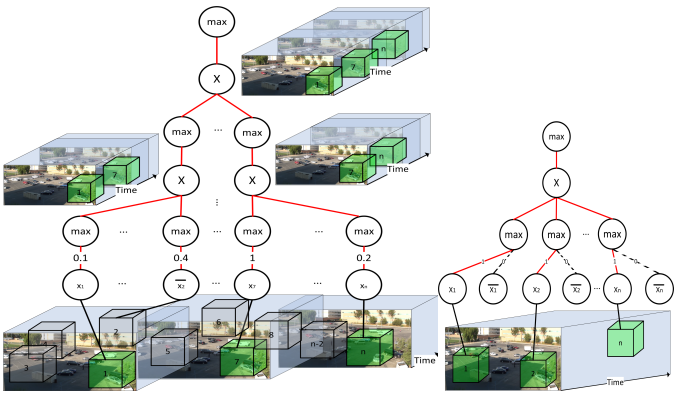
SPN+LR replaces the CG model with the widely popular multinomial logistic regression (LR) for predicting if a space-time window belongs to foreground of a given activity class, $P_{X_b}(X_b = 1 | \boldsymbol{c}_b, y) = \frac{\exp(\boldsymbol{\beta}_y \cdot \boldsymbol{c}_b)}{\sum_y \exp(\boldsymbol{\beta}_y \cdot \boldsymbol{c}_b)}$. SPN+LR uses the same features as our SPN+CG. Note that learning of SPN+LR requires the supervised setting. Maximum likelihood (ML) is used to learn the LR parameters $\boldsymbol{\beta}_y$ in the supervised setting over all space-time windows $\{B_b\}$ labeled as belonging to foreground of class y, $y = 1, \ldots, A$.

CG uses a 3-layered SPN, illustrated in Fig. 4. In CG, all space-time windows are selected as foreground, i.e., $\forall b$, $X_b = 1$ and $\overline{X}_b = 0$, and all windows are connected to their sum nodes at the upper level with weights set to 1. There is only one product node connected to the root. CG is equivalent to the CG model of [3]. CG uses the same features as our SPN+CG. CG parameters are learned in the supervised setting. A comparison of SPN+CG and CG tests the advantages of using a deep model versus the CG model.

Fig. 4 shows the parse graphs of SPN+CG and CG, inferred on an example video from the VIRAT dataset.

8.4 Quantitative Results

Evaluation Metrics include: video classification accuracy, and recall and precision of localizing foreground. A true positive is declared if the intersection of ground-truth and detected foreground is not smaller than 50% of their union.



(a) SPN+CG (b) CG

Fig. 4. Our inference on an example video from the VIRAT dataset: (a) A part of the parse graph using SPN+CG and the inferred foreground (green). (b) CG is equivalent to the counting grid model of [3] which selects all space-time windows as foreground.

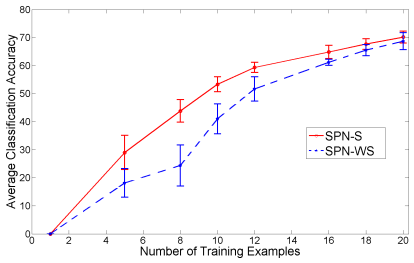


Fig. 5. Average classification accuracy of SPN+CG(WS,V) and SPN+CG(S,V) on the Volleyball dataset as a function of the number of training examples.

Sensitivity to model parameters. Tab. 1 shows sensitivity of SPN+CG to a specific choice of the number of: (a) SPN levels, (b) counting grid points, and (c) grid points enclosed by each space-time window. As can be seen, we are relatively insensitive to (a)–(c) over a certain range of their values. As the SPN height and width increase, the results improve. However, SPN heights above 24 levels, and widths above 10 nodes lead to over fitting. For all our experiments, presented below, we choose the smallest SPN height of 8 levels and width of 10 nodes at non-terminal levels, which give equally good performance as more complex models.

Sensitivity to Number of Training Data. Fig. 5 shows how the number of training examples affects our average classification accuracy on the Volleyball dataset. We examined both learning settings: SPN+CG(WS,V) and SPN+CG(S,V). As can be seen, our performance improves in both settings as the number of training examples increases, and becomes saturated when the number of examples goes above 20. Interestingly, difference in the performance of SPN+CG(WS,V) and SPN+CG(S,V) is relatively small for 20 training examples. This suggests that our approach is able to robustly learn volleyball activity classes from a relatively small number of examples. We observed similar behavior of SPN+CG(WS,V) and SPN+CG(S,V) on the other datasets. For the other datasets, we did not observe overfitting, i.e., decreasing performance for larger numbers of training examples.

Supervision vs. Weak Supervision. Tab. 2 shows that SPN+CG(S) outperforms SPN+CG(WS) in terms of average classification accuracy. This is expected, since SPN+CG(S) has access to additional ground-truth annotations in training. But the differences in their performance range between 1.6% and 3.1% on the KTH, UT-Interactions, VIRAT, and Volleyball datasets. This demonstrates that SPN+CG(WS) successfully relaxes the requirement for expensive manual annotations of foreground in videos. Confusion matrices of SPN+CG(WS) and SPN+CG(S) on the four datasets are shown in Fig. 6.

Tab. 3 presents recall and precision of SPN+CG(WS) and SPN+CG(S) on the UT-Interactions, VIRAT, and Volleyball datasets. Both approaches achieve the highest F-measure when they use a hierarchy of space-time windows with sizes defined by varying $m=\{2,3,4\}$. As expected, SPN+CG(WS) yields worse foreground localization. In some error cases we observed that SPN+CG(WS) identified informative parts of background, providing contextual cues for recognition, as foreground. Considering that SPN+CG(WS) is trained without any access to foreground annotations, its localization performance is quite good in comparison to that of SPN+CG(S).

Comparisons. Tab. 2 shows that SPN+CG outperforms the baselines SPN+CG+Cubes(S), SPN+LR, and CG. In particular, on the Volleyball dataset, accuracy of SPN+CG(S) and SPN+CG(WS) is larger by 12.5% and 10.8% than that of CG, respectively, which quantifies the advantages of grounding SPN onto the counting grid model of [3], even when our deep model is trained under weak supervision. As can be seen, replacing the counting grid model with logistic regression in SPN+LR decreases performance. Also, using the cuboid spatiotemporal features in SPN+CG+Cubes(S) is inferior to our weakly supervised SPN+CG(WS).

Tab. 2 also shows a comparison with prior work: (i) SVM of a Bag-of-Word of SCISA features [36]; (ii) SVM of space-time grids of local features [29]; (iii) SVM with a kernel that accounts for spatiotemporal matches of interest points [41]; (iv) pLSA and LDA models [31]; (v) Convolutional neural networks [37]; and (vi) Action-bank [44]. Interestingly, even without deep learning of local features, SPN+CG+Cubes outperforms the approaches of [29], [31], [36], [37], [41]. The comparison with the action-bank of [44] is unfair to us, hence our lower performance, since the approach of [44] uses a higher level of supervision in training for expressing human activities in terms of simpler actions. Unlike [44], we do not have access to annotations of simpler actions in training.

Valid vs. Invalid Graph Structure. Tab. 4 shows the average classification accuracy, precision and recall of SPN+CG(S,V) and SPN+CG(S,I) on the VIRAT, UT-Interactions, and Volleyball datasets. As can be seen, SPN+CG(S,I) is worse for each evaluation metric. One reason is that the graph connectivity of SPN+CG(S,I)

	SPN height					
Dataset	4	8	16	24	32	64
VIRAT	75.4±1.2	76.2±3.1	76.3±2.1	76.2±1.2	74.6±2.4	70.9±2.6
Volleyball	67.5±2.8	69.8±1.6	69.1±1.3	68±3.1	65±2.3	62.5±2.6

	SPN width			
Dataset	5	10	20	40
VIRAT	68.4±3.7	76.2±3.1	72.7±1.2	70.1±3.8
Volleyball	62.5±2.4	69.8±1.6	66.3±2.8	64.1±2.2

	# of pts in the counting grid is $N \times N \times N$			
Dataset	$N = 16$	$N = 32$	$N = 64$	$N = 128$
VIRAT	63.6±1.4	71.9±3.4	76.2±3.1	74.72±1.5
Volleyball	60.1±1.1	63.6±2.4	69.8±1.6	66.3±2.1

	Window size $(64 \cdot 2^{-m}) \times (64 \cdot 2^{-m}) \times (64 \cdot 2^{-m})$ grid pts			
Dataset	$m = 2$	$m = 3$	$m = 4$	$m = 2, 3, 4$
VIRAT	68.1±3.4	72.72±2.3	74.5±3.7	76.2±3.1
Volleyball	52.5±1.8	61.1±1.4	63.3±2.1	69.8±1.6

TABLE 1

Average classification accuracy in [%] of SPN+CG(WS,V) on the VIRAT and Volleyball datasets using different SPN heights, SPN widths of non-terminal levels, numbers of points in the counting grid, and different sizes of space-time windows. Note that m=$2, 3, 4$ denotes that we use a hierarchy of space-time windows.

Dataset	SPN+CG(S)	SPN+CG(WS)	SPN+CG+Cubes(S)	SPN+LR	CG	[44]	[36]	[37]	[29]	[41]	[31]
KTH	96.8±1	95.2±2.9	94.8±1.6	94.0±1.1	91±1.2	98.2	93.9	90.2	91.8	91.1	83.3

Dataset	SPN+CG(S)	SPN+CG(WS)	SPN+CG+Cubes(S)	SPN+LR	CG	[36]	[14]	[41]
UT	85.5±1.2	82.4±1.3	81.1±1.1	80.3±1.3	75.2±2.3	76.0	75.7	70.8

Dataset	SPN+CG(S)	SPN+CG(WS)	SPN+CG+Cubes(S)	SPN+LR	CG	[36]
VIRAT	78.04±1.2	76.2±1.1	74.1±1.2	72.5±2	70.7±2.2	68.1
Volleyball	71.5±2.1	69.8±2.6	66.5±1.9	65.0±1.3	59.0±2.4	56.6

TABLE 2

Average classification accuracy in [%] on the KTH, UT-Interactions, VIRAT, and Volleyball datasets.

	Window size = $(2^{-m}64) \times (2^{-m}64) \times (2^{-m}64)$ grid points							
	$m = 2$		$m = 3$		$m = 4$		m=$\{2, 3, 4\}$	
Dataset	SPN+CG(S)	SPN+CG(WS)	SPN+CG(S)	SPN+CG(WS)	SPN+CG(S)	SPN+CG(WS)	SPN+CG(S)	SPN+CG(WS)
VIRAT	.97 (.29)	.97 (.18)	.93 (.37)	.90 (.38)	.76 (.54)	.73 (.48)	.72 (.70)	.69 (.64)
UT	.98 (.21)	.96 (.15)	.89 (.26)	.88 (.27)	.75 (.50)	.72 (.46)	.70 (.66)	.62 (.55)
Volleyball	.79 (.28)	.78 (.21)	.72 (.27)	.69 (.24)	.56 (.27)	.52 (.28)	.55 (.51)	.52 (.41)

TABLE 3

Recall and precision (given in the parentheses) for different sizes of space-time windows.

	Classification accuracy		Precision		Recall	
Dataset	SPN+CG(S,V)	SPN+CG(S,I)	SPN+CG(S,V)	SPN+CG(S,I)	SPN+CG(S,V)	SPN+CG(S,I)
VIRAT	78.04±1.2	74.8±2.9	.70	.62	.72	.67
UT	85.5±1.2	81.6±3.8	.66	.60	.70	.66
Volleyball	71.5±2.1	68.3±3.2	.51	.43	.55	.50

TABLE 4

Average classification accuracy, recall, and precision for valid (V) and invalid (I) architectures using SPN+CG(S) with space-time windows of sizes m=$\{2, 3, 4\}$

is manually specified, and violates the completeness constraint. Interestingly, the difference in classification accuracy of the models with valid and invalid graph structures is smaller than the differences in their recall and precision.

Detection of Complex Events. Tab. 5 shows the average precision of SPN+CG in detecting the complex events of TRECVID, and our comparison with the following state-of-the-art approaches. [47] uses the max-margin framework to discover and learn the sequences of latent parts of the complex events. [48] hierarchically combines video features using an AND-OR graph structure, where nodes in the graph represent combinations of different sets of features. [49] fuses multiple features, and learns multi-level relevance labels of training videos. [46] uses linear dynamical system models for capturing the dynamics of time series of windowed mid-level concept detectors. As can be seen, SPN+CG yields the largest mean average precision and thus outperforms the state of the art in detecting complex events "in the wild".

Running time. Without feature extraction, using our MATLAB implementation, inference of SPN+CG on the 4sec Volleyball videos takes on average 11s; and to learn SPN+CG using 20 training Volleyball videos it takes about 700sec. on a 2.66GHz, 3.49GB RAM PC.

8.5 Qualitative Results

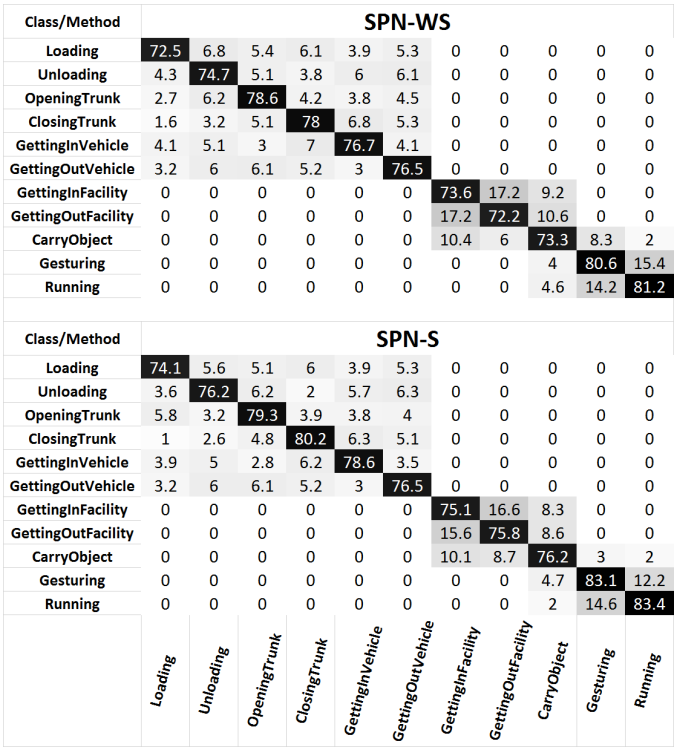
Fig. 7 illustrates foreground localization of SPN+CG on a few frames from example videos of the Volleyball,

Event	Chance	[46]	[47]	[48]	SPN+CG
BT	1.18	**33**	15.44	22.87	29.02
FA	1.06	12	3.55	7.75	**16.82**
LF	0.89	27	14.02	**30.31**	27.98
WC	0.86	31	15.09	38.12	**47.24**
WP	0.93	22	8.17	21.50	**32.58**
Mean AP	0.98	24	11.25	24.11	**40.32**

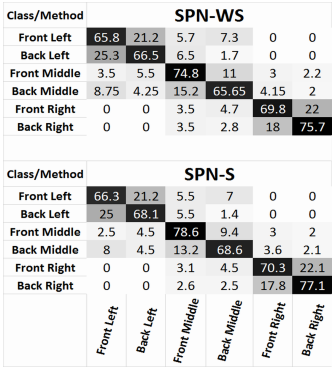
Event	Chance	[49]	[47]	[48]	SPN+CG
BP	.54	16.2	4.38	15.97	**17.02**
CT	.35	25.2	0.92	31.92	**26.12**
FG	.42	54.2	15.29	44.11	**60.04**
VU	.26	**28.2**	2.04	16.32	13.88
GA	.25	9.2	0.74	11.00	**14.34**
MS	.43	10.2	0.84	14.29	**18.08**
PA	.58	**28.7**	4.03	18.53	19.88
PR	.32	**31.2**	3.04	26.10	29.82
RA	.27	29.3	10.88	27.03	**34.76**
SP	.26	18.3	5.48	12.49	**19.24**
Mean AP	.37	25.07	4.77	21.78	**25.22**

TABLE 5

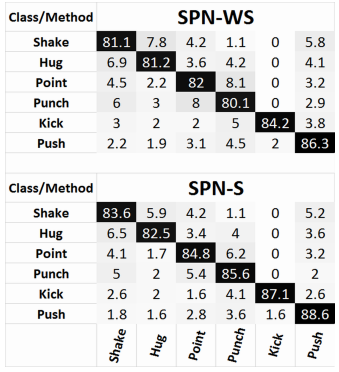
Average precision (AP) in [%] for detection on TRECVID. (Left) DEV-T: "Board trick" (BT), "Feeding an animal" (FA), "Landing a fish" (LF), "Wedding ceremony" (WC), "Wood project" (WP). (Right) DEV-O: "Birthday party" (BP), "Changing a tire" (CT), "Flashmob gathering" (FG), "Getting a vehicle unstuck" (VU), "Grooming an animal" (GA), "Making a sandwich" (MS), "Parade" (PA), "Parkour" (PR), "Repairing an appliance" (RA), and "Sewing project" (SP). The column "Chance" indicates the proportion in [%] of the corresponding event class in the test set.



(a) VIRAT



(b) Volleyball



(c) UT interactions

Fig. 6. Confusion matrices of SPN+CG(WS) and SPN+CG(S) on VIRAT, Volleyball, and UT-interactions.

VIRAT, and UT-Interaction datasets. In particular, the Volleyball video shows the activity "setting-the-ball-to-the-left", the VIRAT video shows the activity "loading-a-vehicle", and the UT-Interactions video shows the activity "hugging". As can be seen, SPN+CG correctly estimated foreground (green windows) in the three videos. Note that in the UT-Interactions video SPN+CG correctly localized "hugging" which simultaneously co-occurs next to the activity "pointing". In the supplemental material, we present two successful example videos from the volleyball dataset. The first video shows the activity "set-to-back-left", and the second video shows the activity "set-to-front-left".

Fig. 8 shows an example video sequence from the UT-Interaction dataset where SPN+CG correctly detected that the video shows the activity "hugging", but also wrongly estimated that the space-time window occupied by the activity "shaking hands" belongs to the foreground of "hugging". SPN+CG got confused because "shaking hands" is very similar to "hugging", and the two activities happen at the same time, near each other. Selecting a smaller size of space-time windows would address this failure example.

Additional examples of our results are presented in the supplemental material.

9 Conclusion

We have addressed video classification and localization of activities with stochastic structure. We have specified a new joint model of sum-product network (SPN) and counting grid (CG) for representing such activities. SPN+CG represents a hierarchical mixture of distributions of counts of visual words, which are detected on a regular space-time grid in the video. New inference and learning algorithms of SPN+CG under both weak supervision and supervision have been formulated. Our inference algorithm has linear complexity in the number of nodes in SPN+CG, when the graph connectivity of SPN+CG is valid. The validity constraint can be ensured by construction.

Due to the valid graph structure, SPN+CG enables exact (i.e., tractable) and thus efficient activity recognition and localization. Our experiments demonstrate that performance of SPN+CG is relatively insensitive over a range of values of model parameters: number of levels, number of nodes in non-terminal levels, number of points in the counting grid, and size of space-time windows. While the weakly supervised SPN+CG yields suboptimal performance relative to the supervised SPN+CG, the former does not need expensive manual annotations of foreground in training videos. The weakly supervised SPN+CG outperforms the state of the art in terms of classification accuracy, and recall and precision of activity detection on the four benchmark datasets: KTH, VIRAT, UT-Interactions, TRECVID MED 2011. We have also compiled a new, challenging dataset of six types of volleyball plays with variable spatiotemporal structures, and small inter-class differences.

ACKNOWLEDGMENTS

This research has been sponsored in part by NSF RI 1302700, and DARPA MSEE FA 8650-11-1-7149.

REFERENCES

[1] S. Oh, A. Hoogs, A. Perera, N. Cuntoor, C.-C. Chen, J. T. Lee, S. Mukherjee, J. Aggarwal, H. Lee, L. Davis, E. Swears, X. Wang, Q. Ji, K. Reddy, M. Shah, C. Vondrick, H. Pirsiavash, D. Ramanan, J. Yuen, A. Torralba, B. Song, A. Fong, A. Roy-Chowdhury, and M. Desai, "A large-scale benchmark dataset for event recognition in surveillance video," in *CVPR*, 2011.

[2] A. Perina and N. Jojic, "Image analysis by counting on a grid," in *CVPR*, 2011.

[3] N. Jojic and A. Perina, "Multidimensional counting grids: Inferring word order from disordered bags of words," in *UAI*, 2011.

[4] H. Poon and P. Domingos, "Sum-product networks: A new deep architecture," in *UAI*, 2011.

[5] R. Gens and P. Domingos, "Discriminative learning of sum-product networks," in *NIPS*, 2012.

[6] M. S. Ryoo and J. K. Aggarwal, "UT-Interaction Dataset," ICPR-SDHA, 2010.

[7] C. Schueldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local SVM approach," in *ICPR*, 2004.

[8] P. Over, G. Awad, M. Michel, J. Fiscus, W. Kraaij, A. F. Smeaton, and G. Quenot, "TRECVID 2011 – an overview of the goals, tasks, data, evaluation mechanisms and metrics," in *Proceedings of TRECVID 2011. NIST, USA*, 2011.

[9] M. R. Amer and S. Todorovic, "Sum-product networks for modeling activities with stochastic structure," in *CVPR*, 2012.

[10] J. M. Chaquet, E. J. Carmona, and A. Fernández-Caballero, "A survey of video datasets for human action and activity recognition," *CVIU*, vol. 117, no. 6, pp. 633 – 659, 2013.

[11] D. Weinland, R. Ronfard, and E. Boyer, "A survey of vision-based methods for action representation, segmentation and recognition," in *CVIU*, vol. 115, 2011, pp. 224–241.

[12] J. Aggarwal and M. Ryoo, "Human activity analysis: A review," *ACM Comput. Surv.*, vol. 43, pp. 16:1–16:43, 2011.

[13] Y. Zhu, N. M. Nayak, and A. K. Roy-Chowdhury, "Context-aware modeling and recognition of activities in video," in *CVPR*, 2013.

[14] M. Amer and S. Todorovic, "A Chains model for localizing group activities in videos," in *ICCV*, 2011.

[15] Z. Zeng and Q. Ji, "Knowledge based activity recognition with Dynamic Bayesian Network," in *ECCV*, 2010.

[16] T. Lan, Y. Wang, W. Yang, S. Robinovitch, and G. Mori, "Discriminative latent models for recognizing contextual group activities," *IEEE TPAMI*, 2012.

[17] T. Lan, Y. Wang, and G. Mori, "Discriminative figure-centric models for joint action localization and recognition," in *ICCV*, 2011.

[18] T. Lan, L. Sigal, and G. Mori, "Social roles in hierarchical models for human activity recognition," in *CVPR*, 2012.

[19] A. Gupta, P. Srinivasan, J. Shi, and L. Davis, "Understanding videos, constructing plots learning a visually grounded storyline model from annotated videos," in *CVPR*, 2009.

[20] Z. Si, M. Pei, B. Yao, and S.-C. Zhu, "Unsupervised learning of event AND-OR grammar and semantics from video," in *ICCV*, 2011.

[21] M. R. Amer, S. Todorovic, A. Fern, and S. Zhu, "Monte Carlo tree search for scheduling activity recognition," in *ICCV*, 2013.

[22] S. D. Tran and L. S. Davis, "Event modeling and recognition using Markov logic networks," in *ECCV*, 2008.

[23] V. I. Morariu and L. S. Davis, "Multi-agent event recognition in structured scenarios." in *CVPR*, 2011.

[24] W. Brendel, A. Fern, and S. Todorovic, "Probabilistic event logic for interval-based event recognition," in *CVPR*, 2011.

[25] S.-C. Zhu and D. Mumford, "A stochastic grammar of images," *Found. Trends. Comput. Graph. Vis.*, vol. 2, no. 4, pp. 259–362, 2006.

[26] L. Zhu, Y. H. Chen, and A. Yuille, "Recursive compositional models for computer vision," *J. Math. Imaging and Vision*, 2011.

[27] M. Pei, Y. Jia, and S.-C. Zhu, "Parsing video events with goal inference and intent prediction," in *ICCV*, 2011.

[28] K. Tu, M. Pavlovskaia, and S.-C. Zhu, "Unsupervised structure learning of stochastic and-or grammars," in *NIPS*, 2013.

[29] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *CVPR*, 2008.

[30] S. Odashima, M. Shimosaka, T. Kaneko, R. Fukui, and T. Sato, "Collective activity localization with contextual spatial pyramid," in *ECCV*, 2012.

[31] J. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised learning of human action categories using spatial-temporal words," *IJCV*, vol. 79, no. 3, pp. 299–318, 2008.

[32] A. Kovashka and K. Grauman, "Learning a hierarchy of discriminative space-time neighborhood features for human action recognition," in *CVPR*, 2010.

[33] X. Wu, D. Xu, L. Duan, and J. Luo, "Action recognition using context and appearance distribution features," in *CVPR*, 2011.

[34] M. M. Ullah, S. N. Parizi, and I. Laptev, "Improving bag-of-features action recognition with non-local cues," in *BMVC*, 2010.

[35] J. Niebles, C.-W. Chen, and L. Fei-Fei, "Modeling temporal structure of decomposable motion segments for activity classification," in *ECCV*, 2010.

[36] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng, "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis," in *CVPR*, 2011.

[37] S. Ji, W. Xu, M. Yang, and K. Yu, "3d convolutional neural networks for human action recognition," *IEEE TPAMI*, vol. 35, no. 1, pp. 221–231, 2013.

[38] L. Liu, L. Shao, and P. Rockett, "Genetic programming-evolved spatio-temporal descriptor for human-action recognition," in *BMVC*, 2012.

[39] M. Baccouche, F. Mamalet, C. Wolf, C. Garcia, and A. Baskurt, "Sequential deep learning for human action recognition," in *Human Behavior Understanding*, 2011.

[40] X. Song, T. Wu, Y. Jia, and S.-C. Zhu, "Discriminatively trained and-or tree models for object detection," in *CVPR*, 2013.

[41] M. S. Ryoo and J. K. Aggarwal, "Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities," in *ICCV*, 2009.

[42] M. Amer, "Volleyball dataset," http://blogs.oregonstate.edu/osupel/dataset/, 2012.

[43] S. Bhattacharya, R. Sukthankar, R. Jin, and M. Shah, "A probabilistic representation for efficient large scale visual recognition tasks," in *CVPR*, 2011.

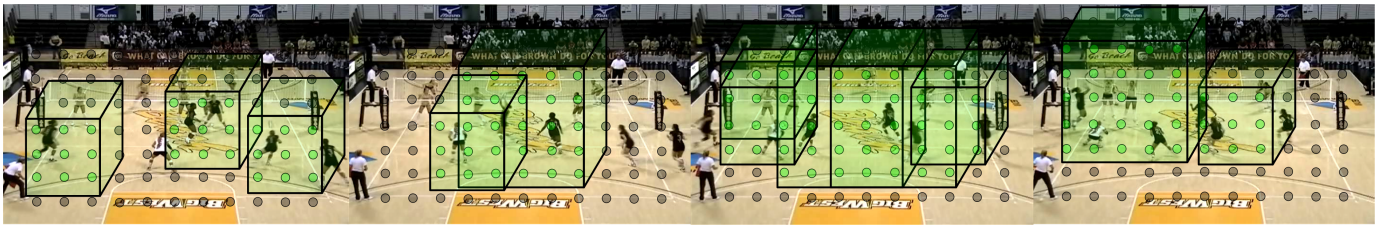
[44] S. Sadanand and J. J. Corso, "Action bank: A high-level representation of activity in video," in *CVPR*, 2012.

[45] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *IWVS-PETS*, 2005.

[46] S. Bhattacharya, M. M. Kalayeh, R. Sukthankar, and M. Shah, "Recognition of complex events: Exploiting temporal dynamics between underlying concepts," in *CVPR*, 2014.

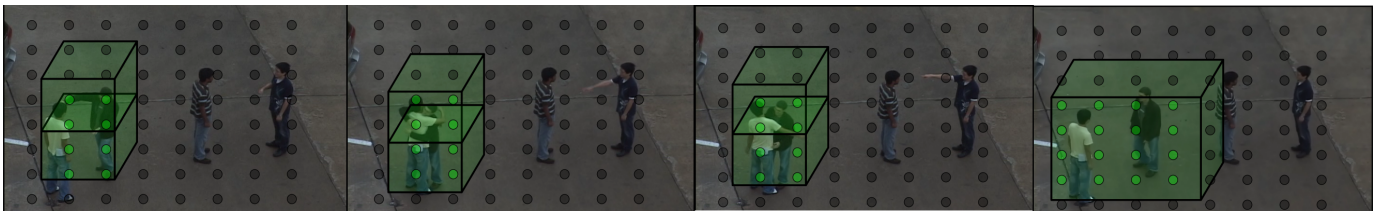
[47] K. Tang, L. Fei-Fei, and D. Koller, "Learning latent temporal structure for complex event detection," in *CVPR*, 2012.

[48] K. Tang, B. Yao, L. Fei-Fei, and D. Koller, "Combining the right features for complex event recognition," in *ICCV*, 2013.



(a) Volleyball Dataset (Set-to-the-left)

(b) Virat Dataset (loading of a vehicle)



(c)UT-Interaction (Hugging)

Fig. 7. Foreground localization of SPN+CG on example vides from: (a) the Volleyball dataset showing the activity "setting the ball to the left"; (b) the VIRAT dataset showing the activity "loading a vehicle"; and (c) the UT-Interaction dataset showing the activity "hugging". The estimated foreground space-time windows are marked green. SPN+CG correctly localized "hugging" which simultaneously co-occurs next to the activity "pointing". We visualize only a subset of points of the counting grid, for clarity.

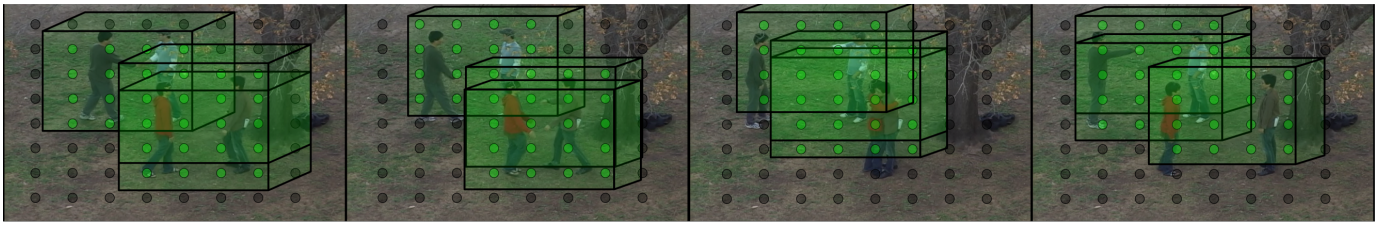


Fig. 8. An example from the UT-Interaction dataset where SPN+CG correctly detected that the video shows the activity "hugging", but also wrongly estimated that the space-time window occupied by the activity "shaking hands" belongs to the foreground of "hugging". SPN+CG got confused because "shaking hands" co-occurs very close to the relatively similar activity "hugging".

[49] Z. Xu, I. W. Tsang, Y. Yan, Z. Ma, and A. G. Hauptmann, "Event detection using multi-level relevance labels and multiple features," in *CVPR*, 2014.

M ohamed R. Amer is a computer scientist at SRI International. He obtained his PhD under the supervision of Prof. Sinisa Todorovic at Oregon State University. He received both his BS and MS at Oregon State University. His main research focus is on affect analysis, multimodal event detection, activity recognition, and tracking.

S inisa Todorovic is an associate professor at Oregon State University. He obtained his MS and PhD degrees in electrical and computer engineering from the University of Florida. He earned his BS/MS in electrical engineering from the University of Belgrade in Serbia. Before going to Oregon State University, Todorovic was a Postdoc in the Beckman Institute at University of Illinois at Urbana-Champaign where he worked on multiscale image segmentation, structural pattern recognition, object / scene recognition and texture classification and synthesis.